The Report Committee for Rohan Sunil Khunte
Certifies that this is the approved version of the following Report:

Proof of Concept for Efficient Compression of Human Vital Signals

APPROVED BY
SUPERVISING COMMITTEE:

Christine Julien, Supervisor

Vijay Garg

Proof of Concept for Efficient Compression of Human Vital Signals

by

Rohan Sunil Khunte

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2019

Dedication

This report is dedicated to my parents, and my sister, without whose support and encouragement I would not have reached this point in my academic and professional endeavors.

Abstract

Proof of Concept for Efficient Compression of Human Vital Signals

Rohan Sunil Khunte, M.S.E.

The University of Texas at Austin, 2019

Supervisor: Christine Julien

The following paper discusses software and data extraction techniques to effectively transmit medical data in its most basic format. The vitals chosen for analysis are the electro-cardiograph (ECG/EKG) signal, and pupillary reflex activity tracking. Between these two vitals, and body temperature, doctors can generally make a large variety of determinations of a person's health. The goal of this paper is to minimize the amount of data that is transmitted, without losing much integrity, and with the hope that it can be sent over very low latency networks, possibly even used for storage in the future. The methods proposed can eventually also be utilized toward a more automated way of detecting irregularities/concerns.

Table of Contents

List of Tables

List of Figures

I. INTRODUCTION

The following report will present techniques that can be used to transmit EKG data, and pupillary light reflex tracking in its most minimal form. Given the advent of remote health/tele-medicine, sometimes data needs to be sent from remote places. For example, as presented in the paper "Telemedicine in Rural India" [1], in a massive country like India, a large percentage of population resides in rural areas whereas doctors tend to be concentrated in the urban areas. Mobile doctors and health units are on the rise, but given the lagging infrastructure, increasing bandwidth efficiency of the networks is of utmost importance. Especially in case of critical failure, or just when vitals are essential to be monitored by a remote healthcare specialist, the need can be dire. Another example of such criticality would be the manned deep space exploration missions of the near future. Each astronaut will have a set of vitals that are sent to Earth so that his/her health is monitored over a large distance. The central theme in these examples is that if more computation can be off-loaded to the front-end and/or back-end, the network or transmission medium doesn't have to carry nearly as much data.

Section II of this paper will focus on the ECG signal. Heart diseases have been on the rise all over the world due to a variety of factors ranging from lifestyle changes to dietary ones. The Centers for Disease Control and Prevention (CDC), according to statistics released for the year 2017, have put the number of adults diagnosed with heart disease at 28.2million, with the number only trending upwards [6]. One of the most powerful tools for non-invasively analyzing the heart's behavior and functions is the ECG device. It is the most widely deployed tool, and with as many as 12-leads, it can generate a lot of data [3]. These leads attach to sensors that are directly placed at strategic points on the thorax of the person undergoing analysis. The output, which is then analyzed on a screen is essentially

just a series of waveforms obtained from filtered electrical impulses. If you observe such waveforms, you will see that it is nothing but a repetitive pattern of a group of three major waveforms, as shown in Figure 1 below, obtained from [5]. As you can see the waveform consists of a P-wave, a QRS-wave (also known as a QRS complex), and a T-wave. Each wave originates from different parts of the heart performing their biological functions of pumping out blood. The P-wave is the electrical impulse from the two atria depolarizing, whereas the QRS complex corresponds to the depolarization of the more muscular ventricles, and T-wave occurs when the myocardium repolarizes. Since the waveforms have been studied and analyzed over a large period, on a diverse and large set of population, the repetitive nature of this waveform, and their corresponding timings can be generalized giving us a better overall idea of when to expect the different datapoints of a regular heart wave. This paper uses these generalizations to prepare algorithms that will detect, analyze, and cut down ECG waveforms to a base statistical file, while not losing a large part of the waveform's integrity.

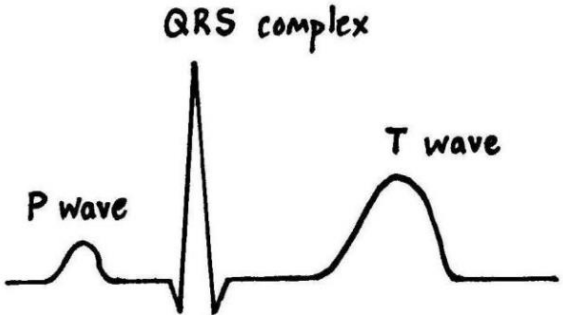

Figure 1: Sample ECG waveform

Section III will then focus on tracking pupillary light reflex. This often-overlooked activity is an important tool to determine brain activity after a high-trauma event. In the United States alone, the CDC statistics show that traumatic brain injury-related Emergency Department Visits, Hospitalizations, and Deaths (EDHDs) have been on the rise [4]. Figure

2 shows this rise, as tracked by the CDC between 2006 and 2014. So, having a quick, automated way to detect, analyze, transmit, and even store pupil reflex information is going to eventually be a consumer need, that technology will have to address. Consider the scenario of high school football in the United States, or similar high impact sports around the world, where there isn't a doctor necessarily present to provide an immediate diagnosis. Instead of walking someone in the field through a refractory test, it would be easy to just have a device that can capture the video of the reaction to light of each eye, compare the behavior, and either make an on-the-spot determination, or send the data to a doctor for immediate analysis and determination. Instead of sending the entire video or multiple images, this paper will propose a way to analyze the data so as to only send the bare minimum information, which can even be used to recreate the original reaction of the pupil using computer generated graphics.

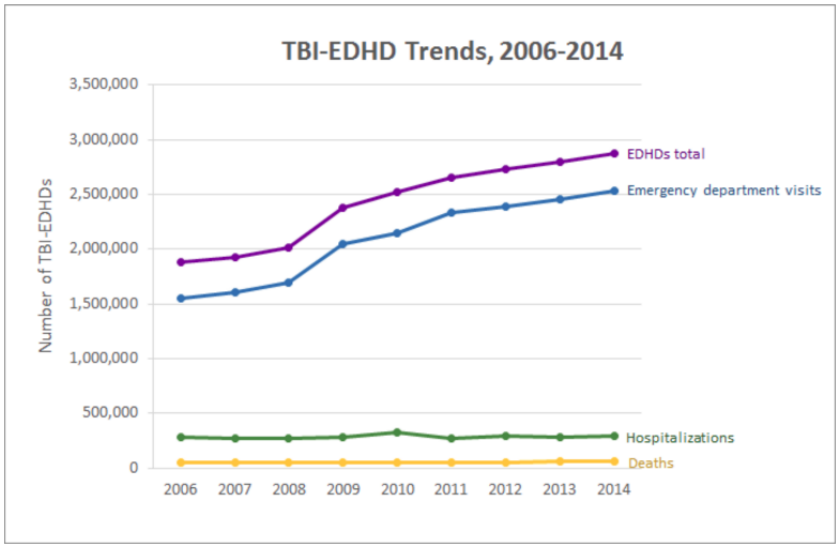

Figure 2: Traumatic Brain Injury EDHD increase over the years

Just to provide a quick background of what exactly the doctors look at, Figure 3 shows a normal eye, which can be compared to Figure 4, that shows a dilated eye. A

person's eyes are dilated when they are: suffering from brain injury/disease, under effects of certain medication/recreational drugs, or sexually attracted to someone. However, this is not the natural state, and hence, when a doctor is examining someone with dilated pupils, they shine a light into each eye to see whether the pupil contracts due to the light passing into the eyes. The shining light essentially triggers the optic nerve to send a signal to the mid-brain, which in turn activates the ciliary ganglia that control the muscles which help the pupils to contract. This shows that for a seemingly simple movement, there are a lot of places for failure to strike, and so, the reaction of the pupil can be very revealing to a doctor.

Figure 3: Normal Pupil

Figure 4: Dilated Pupil

Section IV of this paper provides some concluding remarks regarding the observations made.

II. ECG DATA

Extraction Scheme:

Utilizing a top-down approach, with the final goal of re-creating an accurate ECG signal using statistical data from the original, the important parts of the waveform were found to be:

a. Determining wave crests (P,QRS, and T), and their relation to the mean values of the graph

b. Determining each distance in the P, QRS, and T wave sub-system

c. Detecting distance between consecutive crests for each (example between two P-waves, two R-waves, etc.)

d. Spurious/abnormal data recorded separately

To help with the same, different mathematical operators will be used as data is parsed in a fixed window of step-sizes. The code essentially traverses the dataset, and calculates a travelling mean which is used to determine deviation, and hence track the peaks or troughs. A lot of papers have proposed different algorithms for the detection of such waveforms: like in the paper by Friganovic, et al [10], by Nabil D. and Bereksi-Reguig F. [11], and by Panigraphy D. and Sahu P. K. [12]. When studied closely, each paper seemed to be a variation of moving averages run on fixed-window widths, and on filtered datasets. The Friganovic, et al paper [10] was especially instrumental in selecting the method of detecting the three waves that is used in this paper. The Nabil and Bereksi-Reguig paper is especially useful in automating the ways in which one can detect the widely different morphologies presented by abnormal heart beats. On another hand, the method proposed by Panigraphy, et al. [12] also contributed in terms of getting a closer on/off location for the P and T-waves. However, the method selected to detect the QRS wave in this report was presented

by Mohamed Elgendi [13], and it was refined by him and others, to detect the P and T waves [14]. As we move forward in the paper, the first method will be referred to as the "Elgendi" method, and the follow-up method to be the "Elgendi revised" method. According to these methods, assuming a clean ECG signal, parsing windows are created with pre-determined wave widths, and pre-determined variation from the running average. In this paper, the slope is set to +/- 10mV per sample for the R wave, and +/-6 mV/sample for all other waves. The time-scale window widths are based on empirical data, and are summarized in Table 1 below. A note to bear in mind here, is that the T-wave is the most volatile and non-uniform wave of the three, making it the hardest to estimate. For this reason, we have assumed a width of 100ms with a slightly higher tolerance. The Q-T transition should also be carefully considered for a higher tolerance, as it is mostly a function of the R-R frequency, and can vary from person-to-person. As each respective window is parsing, we first apply a regular band-pass filter to remove noise from the 0.5Hz to 10Hz range, as suggested by Elgendi in [13]. Lastly, abnormal data, like those in the case of arrythmia or other heart conditions will be recorded as separate events for this paper.

Subject	**Average width (in milliseconds)**	**Tolerance (in milliseconds)**
P wave	110	+/-20
P-R transition	190	+/-30
Q-R-S complex	110	+/-20
Q-T transition	300	+/-110
T wave	100	+/-40

Table 1: Average ECG readings in healthy individuals [5][10][13]

The algorithms, when combined, return indices of where particular data is located. Using this information, we will track the following key information, which correspond to important areas of the wave, as shown in Figure 5; ms: milliseconds, mv: millivolts below:

- Time-stamp of first P-wave (ms): Start-locate
- Average width of P-wave (ms): P-t
- Average height of P-wave(mv): P-v
- Average peak width of P-wave: P-peak
- Average distance from P-wave to Q-wave: PQ-t
- Average width of the QRS-complex (ms): QRS-t
- Average height of Q-wave: Q-v
- Average height of R-wave: R-v
- Average peak width of R-wave: R-peak
- Average height of S-wave: S-v
- Average width of the S-T transition: ST-t
- Average width of the T-wave: T-t
- Average height of the T-wave: T-v
- Average peak width of T-wave: T-peak
- Time-stamp value of last processed t-wave: Stop-locate
- Average dead-zone distance: D-t
- Average dead-zone height: D-v
- Spurious wave [n] start time: Spur[n]-Start-locate
- Spurious wave [n] width: Spur[n]-t
- Spurious wave [n] height: Spur[n]-v

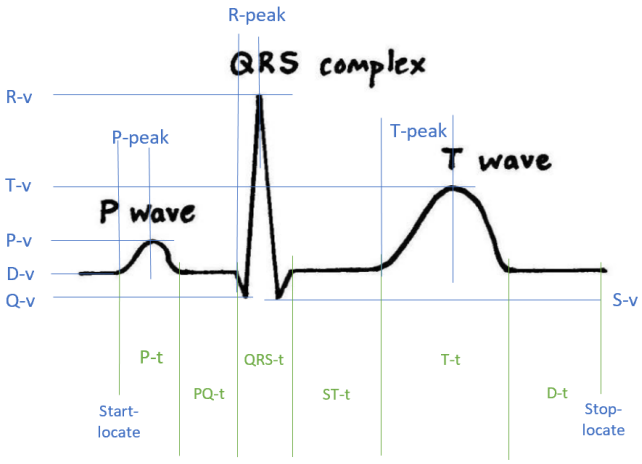

Figure 5: Key information mapped to ECG wave

After this, the algorithm indicated by the block-diagram in Figure 6 below is followed:

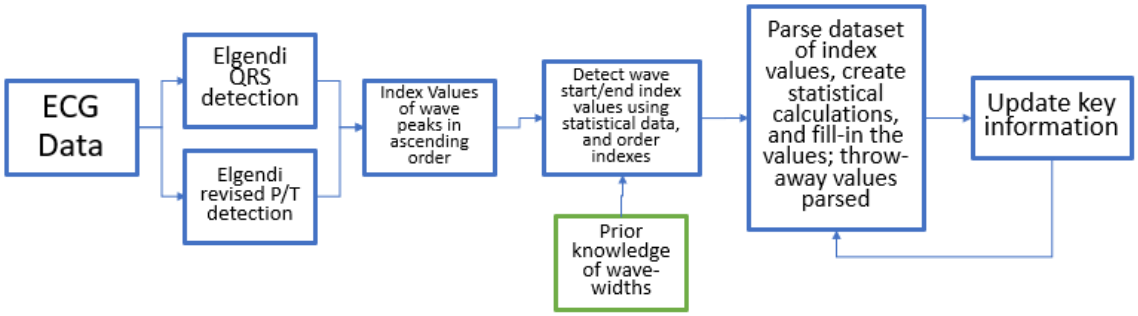

Figure 6: Algorithmic Flow to analyze ECG data

The height averages above will be calculated only if the corresponding reported height is within a +/- 4 to 6mV (or +/- 8 to 10mV in case of the R-waves) margin of the average values in the optimized dead-zone. The reason for doing so is to not have an unnecessary train of spurious wave data, while at the same time not ignoring these waves. An example of this would be in the case of arrythmia, the T-wave on some of the lead can be levels of magnitude away from. The width averages above will be calculated based on

the indices marking the start and end of the waves. The number of indices will be converted into a millisecond value based on the sampling rate used. In the end, the code will just spit out key information, which we can use to plot a brand-new ECG wave. In the future, machine learning algorithms can be applied to a collection of such statistics to develop patterns.

So how will the plotter work? As shown in Figure 5, we will always start at time zero, and will plot values equal to D-v till time Start-Locate, where we will start the P wave plotting with a curved line going from value D-v to value P-v, based on a weighted location close to either ends of the wave. This is followed by plotting the D-v value for time of PQ-t. Then we repeat the same procedure for the QRS-complex, and T-wave, and then continue unless plotter time equals the start location of one of the spurs. In which case, a spur is plotted and then the width of the spurious wave is removed from the D-t, to allow the next P-wave to be plotted.

Experimental Analysis:

Since working with real-time data would require a lot of front-end hardware work, this paper utilized ECG data published online by the PhysioBank database [9], sampled at 720Hz with a 12-bit bandwidth. The ".dat" format was first converted to txt, and then to a csv for the ease of parsing through in the code, which is done as a backed server model, in the "nodejs" language. For the purposes of this paper, we will assume that the ECG waveform obtained will be close to the ideal, because spurious data was not handled well, as explained later. When run initially, it was realized that without weighting, the P-wave and the T-wave won't be accurately represented by the plotter. Hence, the peak location width value was added to the key data field. This way, a lot more integrity is added, with a little overload.

When the key data was obtained from the first dataset, and eventually recreated, the plotter continued to be off with regards to timing, with an error of about 8-10ms, as we will see in Figures 7 and 8 below. This can possibly be attributed to the fact that once the algorithm offloads indices, the code goes in and calculates the start and end points using pre-defined lengths, and averages. A glaring example of this can be seen in the case of the S-wave. Since the R-position was known from the Elgendi method, and the QRS-complex length was assumed to be known (from Table 1 above), the S-v value was supposed to be anywhere between a certain fixed width of samples/time. Since the threshold for detection was about +/- 6mV deviation, the first high value away from the D-v was assumed to be the S-v. This value also ended up being much later in time, and hence a greater slope on the R-S side of the wave. Another factor contributing to the error could also be the fact that indices obtained from the Elgendi method, being translated to a time value can vary from the original value due to smoothing effects of the band pass filters used.

Having said that, we can see in Figures 7 and 8 that the two graphs track each other pretty closely in most parameters that matter. Figure 7 is the raw output as downloaded from the databank. Figure 8 is the Frankenstein image that the code generated when everything is calculated linearly. If the looks are ignored, the timing, and ECG waveform voltage values seem to be perfectly in-sync. If we add a smoothing sinusoidal function to the P and T waves, we will get a better-looking graph, reminiscent of Figure 5.

Figure 7: Raw Waveform Plot of aami3a

Figure 8: Plotted Waveform from Key Information with Linear Math

Another glaring assumption that is challenged, is our PQ transition time of 190 +/- 30ms. On an average, we are seeing times greater than 400ms. Upon further reading it is highly possible that aami3a data chosen was for a person suffering from heart blockage of some sort. Such elongated lengths are a major medical condition, and from the looks of the key information, we were able to extract it nicely.

Lastly, this paper has chosen to omit spurious data for now, as more refinement is needed to handle the waveform accurately. From the representation above, this algorithm

needs a lot of modification for statistical spur tracking, if it is indeed going to work on complex and real-time ECG signals, the work of which lies outside the scope of this paper. However, the central theme of this paper was to explore whether large files of data are the only essential way to represent an ECG signal. As shown in Table 2 below, it is not. Using "key data" what we have done is created a file, which within a few kilobytes of space, stores more than enough information to generate a waveform with enough accuracy to transport ideal ECG data. The comparison below is done with the ".dat" format used for compression by the MIT lab, as well as the raw csv file size.

Sample Filename	**.csv size**	**.dat size**	**.txt key info size**
aami3a	210 KB	84 KB	210B
aami3d	169 KB	67 KB	210B
aami4b_h	319 KB	127 KB	211B

Table 2: File size comparison for ECG waveforms

III. PUPIL REFLEX TRACKING

Extraction Scheme:

The central theme in this data vital is to check if and how the pupils of the eye react to light. As explained before, the shining of light makes the pupils contract as a response. With a closeup video of the eye as shown by YouTube user Dr. A Asha [15], analysis becomes much easier, and we can get away with basic gray scaling, edge-detection, and shape-detection algorithms. Using a top-down approach like the last section, let's start with key information that will help us reconstruct the exact scenario that the doctor is looking for:

- Radius of pupil before contraction:R-b
- Radius of pupil after contraction:R-a
- Time required for transition(ms):T
- Color of the iris:Color

And optionally, we could also try and detect the percentage of redness of the sclera if the image is clear enough. Figure 9 illustrates the basic elements of the outer eye to get an idea of the basic anatomy, followed by Figure 10, which proposes an extraction algorithm for getting our key information.

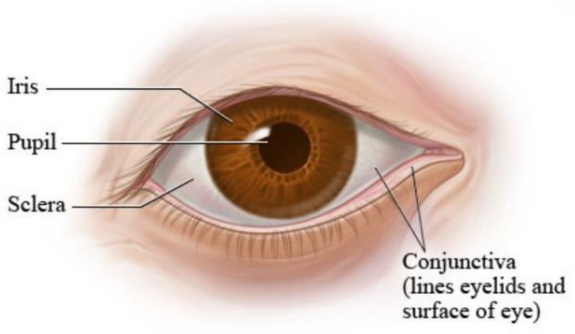

Figure 9: Basic visible outer elements of the eye [16]

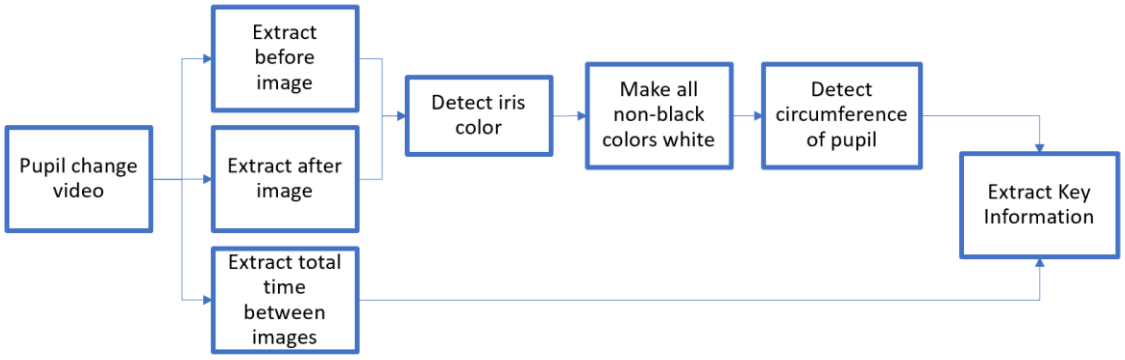

Figure 10: Algorithmic Flow for Pupil Reaction Extraction

There are various algorithms available for each stage above. Let's consider the case of edge detection. As researched by Afshin Ghanizadeh, et al. [17], a side-by-side comparison of the Sobel, Canny, and their own proposed edge detector, surmised by them in the Figure 11 below, edge detection algorithms have reached a level where the application can now select how much it really wants to detect, based on further processing requirements. Evident below is that the three filters work great in edge detection, and can get us as much detail as desired. Although the authors' proposed filter works well to filter out the unnecessary information, similar performance, or avoidance of features can be obtained from the Canny edge detection algorithms available freely, and widely used (in this case, OpenCV in Python). If we control the filter parameters, and restrict the edge pixel comparison, we can get an adequate amount of useful data. Furthermore, we plan to be zoomed-in on the eye more than the pictures shown in Figure 11, and we will also convert any other non-black color to white using a thresholding function. This will offload a large portion of complexity from the edge detection algorithm, making it much less resource-intensive.

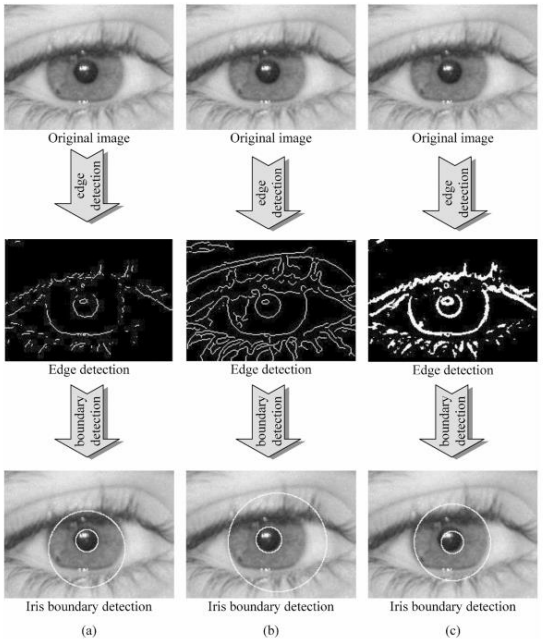

Figure 11: Comparison of edge detection algorithms shown in [18]

The next choice to make is detection of the radius. Fortunately, contour detection is much easier in the binary world (which is why we have this edge-detected image). However, shape detection algorithms available openly work only on basic shapes. Luckily, we are just detecting a black circle in our case, and there are a variety of ways this can be done, such as, using the Circle Hough Transform (CHT) as presented by Ali Ajdari Rad, et al [20], which will be used in this paper for its one-function implementation of the algorithm which directly gives us our desired results. Among other options considered by this paper for improving efficiency, was the use of gradient pair vectors similar to the concept of edge detection, but for matching patterns, as suggested by "The Adaptive Hough Transform" paper [21]. However, the more options that were investigated, it seemed like the proposals were just modifications to the HCT algorithm, with a focus on reducing time and power utilized by the same. The underlying ideas of each traces back to the HCT concept that: when a circle of an unknown radius exists, we traverse through multiple radii

value, and try to find a set of circles which solve the equations: $x=a+R*\cos(\Theta)$ and $y=b+R*\sin(\Theta)$ where (a,b) is the center of the circle, and r is the radius by going through all "edge detected" points x,y available in the image. When such circles are located, they are represented simply as an array of [(a,b),R]. The computational power required by this algorithm is quite large since we are solving an array of three-dimensional models. However, given that our input is going to be once circle detected by a very strong edge, the Hough transform is an ideal start point. And the fact that this will directly return a radius value for both images, makes it an easy choice to obtain our key information.

Using the differences in radii, we can then measure the reaction of the pupil reflex, or in the case of traumatic brain injury, lack of any pupil reflex. This information, although vital within itself, would also make it possible to later generate an image of an eye using computer generated graphics (not in the scope of this paper). In fact, the entire video can be approximated to recreate the pupillary reflex movement, and as a bonus, include the correct color of the iris of the eye (also not in the scope).

Experimental Analysis:

Since there are no databanks for showing pupillary reflex activity, we will have to make do with those available online freely on websites like YouTube. So, for this experimentation our data is going to be a set of one video [19], with Python being the chosen language so that more open source utilities can be used. As presented in the previous section, the algorithm first calls for extraction of the before and after images. Since the video is a zoomed-out view of a reaction, there is a lot of color noise. Extraction, especially of the time, but also the images, would be magnitudes easier with more control of the environment, camera exposure/angle/distance. For this purpose, a part of the video is extracted manually using video editing software, and then the first frame and last frame are

treated as the first image and last image respectively, and the period of the video, treated as the time between images.

For detection of color, there were a variety of complex options available, but the method chosen by this algorithm correctly detected the RGB ratios corresponding to the amber iris color. Basically, color detection was achieved by creating RGB tuples to act as upper and lower bounds of the color to be detected. Then the code parsed through the image, and masked over the detected color. Just to try this on a few other iris colors, a blue eye, and a green eye was also tested, and detected.

The next step calls for detecting the edges of the pupil, so that the radius can be obtained, and stored. In actuality, as discussed in the previous section, this is technically the first step in the goal of contour/circle detection. Once the edges are procured and the entire image is binary, with only data that is important to you, contour detection can be achieved in a single line of code. With this end in mind, the open-source canny edge detection function available via OpenCV was used for this analysis. When the edge detection image was observed in the first go, there were a lot of unwanted edges detected because of the different colors, and so, the decision was made to first convert the image to grayscale, and then using thresholding techniques to white out as much of the gray matter as possible. Edge detection of this new image was much easier, as shown in Figures 12 and 13.

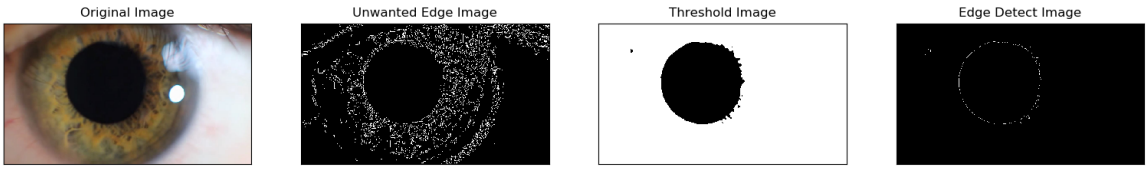

Figure 12: “Before” Image showing a Dilated Pupil

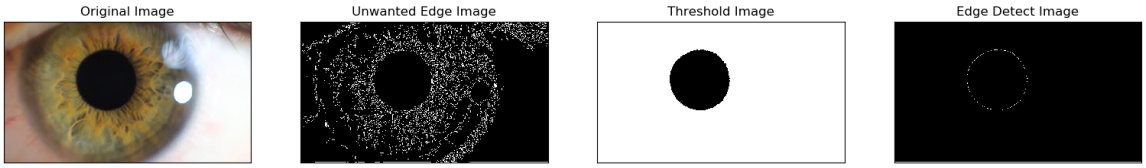

Figure 13: “After” Image showing the Pupillary Reflex to Light

These edge-detect images were then run through the circle detection HCT algorithm, and we were able to successfully detect our pupil as shown in Figure 14. As explained in the algorithm, various radii values were tried, and the one that worked was returned. We repeat this process for the “after” image, and are ready with our key information, which is a file of a mere 39 bytes, as compared to the video that can take up as much as 2.68MB in mp4 format.

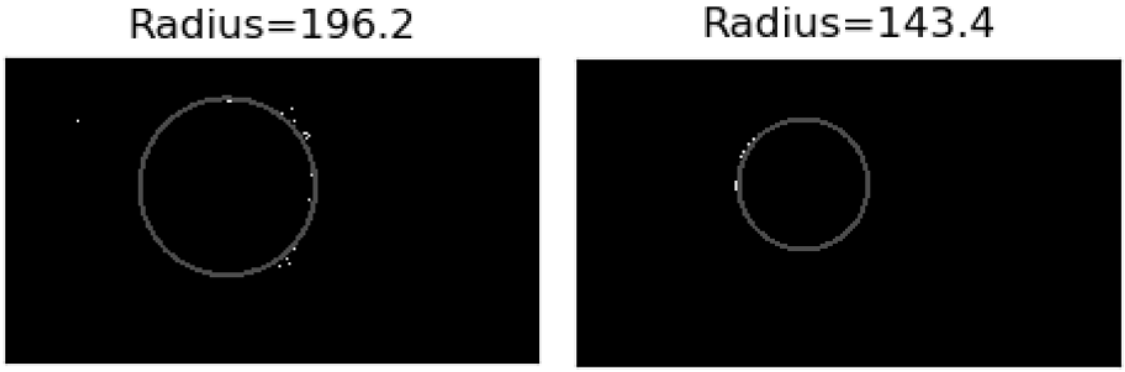

Figure 14: Contour detection on the Edge Detected Pupil, before and after

Word of caution here: since the distance to the eye from the camera lens is not known it is impossible to determine what units we should be using for the radius, except number of pixels. The difference can indeed give us a volume of change, but unless we use standardized values, there is no way to concretely say that an image recreated using this data is an exact size approximation of the original. However, for the purposes of simply tracking the reaction, this is a great tool.

IV. CONCLUSION

This paper has shown conclusively that with prior data, and deterministic environments, it is indeed possible to shrink human ECG and pupillary reflex information to statistical values and/or measurements, that we call “key data/information.” The proof of concept introduced here has been mostly accomplished in a noise-free, and ideal database environment. Translation to the real world will require a lot of processing power, storage, and cache memory on the front data-collection end. With an increasing rate of investment in edge-detection ICs, and high-powered embedded CPUs, the requirements for getting key data are realistic. For those interested in pursuing this goal further, a variety of options are available on the front-end side (to handle spurs, and reduce complexity/power-usage), and on the back-end side (to try and make the graphs appear aesthetically similar to a direct ECG wave. Especially in the case of ECG, if a waveform changes rapidly, the current implementation keeps generating spurs, which is not viable. A recommendation to combat this would be increasing the size of our key data. Since thousands of kilo-bytes of data is already being saved, this increase would still be acceptable.

To surmise, ECG and pupillary reflex data, can be converted from their current large-data format (numbers and video respectively) to basic set of fixed numerical values which, upon analysis, yield a terrific amount of diagnosable information.

References

[1] Bagchi, S. (2006) Telemedicine in Rural India. PLoS Med 3(3): e82. https://doi.org/10.1371/journal.pmed.0030082

[2] Wong, P. (2017). ECG Signal Compression and Optimization in Remote Monitoring Networks.

[3] Lome, Steven. 10 Steps to Learn ECG Interpretation. https://www.healio.com/cardiology/learn-the-heart/blogs/10-steps--course-to-learn-ecg-interpretation

[4] United States of America, Centers for Disease Control and Prevention. National Center for Injury Prevention and Control. https://www.cdc.gov/traumaticbraininjury/data/tbi-edhd.html

[5] Price D. South Sudan Medical Journal Vol 3 Issue 3 May 2010. http://www.southsudanmedicaljournal.com/assets/files/Journals/vol_3_iss_2_may_10/ECG%20article.pdf

[6] United States of America, Centers for Disease Control and Prevention. Summary Health Statistics: National Health Interview Survey, 2017. Table A-1a. https://ftp.cdc.gov/pub/Health_Statistics/NCHS/NHIS/SHS/2017_SHS_Table_A-1.pdf

[7] Swarup S, Makaryus AN. Digital stethoscope: technology update. *Med Devices (Auckl)*. 2018;11:29–36. Published 2018 Jan 4. doi:10.2147/MDER.S135882

[8] World Health Organization Cardiovascular diseases (CVDs) [Accessed 20th April 2019]. Available from: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[9] PhysioBank Databases [Accessed 28th February 2019]. Available from: https://physionet.org/physiobank/database/

[10] Friganovic K, et al. Optimizing the Detection of Characteristic Waves in ECG Based on Processing Methods Combinations. Available from: https://ieeexplore.ieee.org/document/8463463

[11] Nabil D. and Bereksi-Reguig F. Algorithm for Automatic Detection of ECG Waves. Available from: https://doi.org/10.1142/S0219519410003848

[12] Panigrahy, D. and Sahu, P. K. Australasian Physical & Engineering Sciences in Medicine. *P and T wave detection and delineation of ECG signal using*

differential evolution (DE) optimization strategy. Available from: https://doi.org/10.1007/s13246-018-0629-8

[13] M. Elgendi, "Fast QRS detection with an optimized knowledge-based method: Evaluation on 11 standard ECG databases", PLoS ONE, vol. 8, no. 9, pp. e73557, 2013.

[14] M. Elgendi, M. Meo, D. Abbott, "A proof-of-concept study: Simple and effective detection of P and T waves in arrhythmic ECG signals," Bioengineering, vol. 3, no. 4, pp. 26, 2016.

[15] Dr. A. Asha. "Abnormal - Pupillary Light Reflex," YouTube [Published 21st April 2009], Available at: https://www.youtube.com/watch?v=sBslzAnFZ7A

[16] Healthwise Staff, My Health, Government of Alberta, Canada. "Eye Structures, Front and Side Views," [Current as of 17th July 2018], Available at: https://myhealth.alberta.ca/Health/pages/conditions.aspx?hwid=tp9807.

[17] Afshin Ghanizadeh, Amir Atapour Abarghouei, Saman Sinaie, Puteh Saad, and Siti Mariyam Shamsuddin, "Iris segmentation using an edge detector based on fuzzy sets theory and cellular learning automata," Appl. Opt. 50, 3191-3200 (2011).

[18] Afshin Ghanizadeh, Amir Atapour Abarghouei, Saman Sinaie, Puteh Saad, Siti Mariyam Shamsuddin, "Iris segmentation using an edge detector based on fuzzy sets theory and cellular learning automata," Appl. Opt. 50, 3191-3200 (2011); https://www.osapublishing.org/ao/abstract.cfm?URI=ao-50-19-3191.

[19] Tjado. "Close up eye and pupil dilation", YouTube [Published 1st March 2012], Available at: https://www.youtube.com/watch?v=DW2iwEshWME.

[20] Ali Ajdari Rad, Karim Faez, Navid Qaragozlou, "Fast Circle Detection Using Gradient Pair Vectors," Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney.

[21] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," PAMI-9 , Issue: 5, 1987, pp 690-698.